

## REMARKS

### **I. INTRODUCTION**

Claims 1-9, and 11-18 remain pending in the present application. No new matter has been added. In view of the following remarks, it is respectfully submitted that all of the presently pending claims are allowable.

### **II. THE EXAMINER HAS OBJECTED TO THE DRAWINGS**

Applicant has filed replacement drawings with this response addressing the Examiner's objection. Thus, the objection to the drawings should be withdrawn.

### **III. THE 35 U.S.C. § 112 REJECTIONS SHOULD BE WITHDRAWN**

Claims 2 and 12 stand rejected under 35 U.S.C. § 112, second paragraph, as failing to set forth the subject matter which Applicants regard as their invention. (*See Office Action, p. 2, ¶ 2*). Applicants respectfully submit that claims 2 and 12, as written, particularly point out and distinctly claim the present invention. The Examiner states that claim 2 fails to correspond in scope with that which Applicants regards as the invention because of the phrase "automatically generates the first code" of claim 2, which depends on claim 1. The Examiner correctly states claim 1 indicates that the first code already exists. However, the Examiner should note that dependent claim 2 adds the limitation of a method for generating the first code referenced in claim 1. As noted in the preamble of claim 2, "the method according to claim 1, wherein the code is generated according to the following substeps..." It is clear from the claims

that while claim 1 indicates a first code exists, claim 2 simply provides the method of generating the first code. Thus, it is respectfully submitted that the objection to claim 2 should be withdrawn. Claim 12 recites limitations substantially similar to those of claim 2, including “automatically generates the first code.” Thus, it is respectfully submitted that claim 12 is allowable for at least the same reasons stated above with reference to claim 2. In view of the remarks above, it is respectfully submitted that the rejections under 35 U.S.C. § 112, second paragraph, be withdrawn.

#### **IV. THE 35 U.S.C. § 102(e) REJECTIONS SHOULD BE WITHDRAWN**

Claims 1-9, and 11-18 stand rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,493,868 to DaSilva et al. (hereinafter “DaSilva”).

DaSilva generally describes an integrated code development tool for developing and analyzing embedded real-time software. This code development tool enables a developer to configure, build, debug, trace, and analyze applications for digital signal processor (DSP) chips. (*See* DaSilva, col. 1, ll. 38-43). The integrated development tool includes a set of application program interface (API) calls for real-time tasks. (*See* Id., col. 1, ll. 43-44). The integrated development tool further includes: DSP Code Generation Tools; Integrated Development Environment; and DSP/BIOS plug-ins and API calls. (*See* Id., col. 1, ll. 50-54). The DSP Code Generating Tools provides the environment for the development and production of DSP assembly language source code from compiled C source code. (*See* Id., col. 1, ll. 56-65). The Integrated Development Environment (IDE) is designed to edit, build, and debug DSP target programs. (*See* Id., col. 2, ll. 25-28). The DSP/BIOS plug-ins and API calls provide support for

real-time program analysis, allowing the developer to probe, trace, and monitor an application. (See Id., col. 2, ll. 62-67).

With respect to editing, the IDE allows the developer to “edit C and assembly source code” and to “view C source code with the equivalent assembly instructions shown after the C statements.” (See Id., col. 2, ll. 30-33). The C source code with the equivalent assembly instructions shown after the C statements is viewable to the developer via an editor window. (See Id., Fig. 4 and col. 1, ll. 23-25). Within the IDE, a Configuration Tool may be used to create files that define objects used by the DSP/BIOS plug-ins and API calls. (See Id., col. 3, ll. 23-25). More specifically, the Configuration Tool sets the parameters used by the DSP/BIOS API calls at run-time, and it serves as a visual editor for creating and setting properties for run-time objects. (See Id., col. 3, ll. 29-33).

The present invention relates to representation of data/file codes (or command nodes) using a command structure. Claim 1 of the present invention recites a method executed on a computing device to perform an operation on extracted elements of a first software code comprising the steps of “generating a list of desired elements of the first software code” and “extracting the desired elements from the first code” in combination with “performing an operation on the extracted elements.” The Examiner contends that the limitation of “generating a list of desired elements of a first software code” is anticipated by DaSilva’s illustration in Fig. 4. (See Office Action, p. 4, ¶ 2). As described in the Applicants’ disclosure, the list to be generated is a predetermined list that includes a plurality of desired elements of the command node that will be extracted. (See Specification, Fig. 15 and p. 26, ¶¶ 0044-0045). Once an element is extracted, it is reviewed to determine if such element is one of the listed predetermined elements, (See Id.)

In contrast, the illustration in Fig. 4 represents “an editor window for viewing C source code with the equivalent assembly instructions shown after the C statements.” (*See DaSilva, col. 1, ll. 23-25*). The editor window depicted in Fig. 4 simply provides a display of the C source code with the assembly source code (compiled C source code) to the developer. (*See DaSilva, col. 2, ll. 31-33*). A list of code may be displayed through the editor window of DaSilva , however the list in Fig. 4 does not teach nor suggest a list of desired elements of the command node that will be extracted. Specifically, the illustration in Fig. 4 does not distinguish between which elements of the code are desired elements on a predetermined list; it merely provides a list of compiled code along side the original code. Therefore the editor window of Fig. 4 does not serve an analogous function to the method of generating a predetermined list of desired elements that will be extracted from software code of the present invention. Thus, it is respectfully submitted that DaSilva’s disclosure neither teaches nor suggests “generating a list of desired elements of a first software code” as recited in claim 1.

Additionally, the Examiner contends that the limitation of “extracting the desired elements from the first code” is anticipated by DaSilva’s description of the Integrated Development Environment. (*See Office Action, p. 4, ¶ 5*). As discussed above, a desired element is an element of the command node that may be included on the generated list of predetermined elements to be extracted from the command node. A review of the extracted element from the command node is performed in order to determine if such element is one of the predetermined (desired) elements on the list. (*See Specification, Fig. 15 and p.26, ¶¶ 0044*). None of the features of DaSilva describe an equivalent method to extracting an element from a source code and reviewing the extracted element to determine if such element is on the list of the

predetermined elements. The integrated editor of the IDE of DaSilva only provides for a limited number of features such as: highlighting words in color, marking blocks of C source code to find the next matching block, changing indentation and tabs, searching within one or more files, using multiple windows, and providing help. (*See* DaSilva, col. 2, ll. 30-41). These features allow for adjustments in the visual representation of C source code on the editor window, however the features do not serve the same function as extracting an element from a command node and determining if such element is on a list of desired element. Thus, it is respectfully submitted that DaSilva's disclosure neither teaches nor suggests "extracting the desired elements from the first code" as recited in claim 1.

Applicants respectfully submit that for at least the reasons stated above, claim 1 of the present application is not anticipated by DaSilva, and request that the rejection of this claim be withdrawn. As claims 2-9 depend from, and therefore include all the limitations of claim 1, it is hereby submitted that these claims are also allowable.

Claim 11 recites limitations substantially similar to those of claim 1. Such limitations include "a first engine receiving a list of desired elements of a first software code," and "a second engine extracting the desired elements from the first code." Therefore, Applicants respectfully submit that claim 11 is allowable for at least the reasons discussed above with regard to claim 1. Because claims 12-18 depend from, and therefore include all the limitations of claim 11, it is hereby submitted that these claims are also allowable.

## **CONCLUSION**

In light of the foregoing, Applicants respectfully submit that all of the now pending claims are in condition for allowance. All issues raised by the Examiner having been addressed. An early and favorable action on the merits is earnestly solicited.

Respectfully submitted,

By:   
Michael J. Marcin (Reg. No. 48,198)

Fay Kaplun & Marcin, LLP  
150 Broadway, Suite 702  
New York, NY 10038

Dated: October 27, 2005